Unexpected Costs—Lessons Learned

Written by an experienced remote payload user and edited by the POIC staff.

Please note that the described experiences are from the early days of POIC and TreK support, and many of the described situations no longer apply. This document is an excellent description of how an evolving program can result in unforeseen additional costs.

Computer Investment

History

We started believing a remote payload would only need one computer for flight support. We opted to double that into two computers, so there would be a "hot swap" computer for support, should the first fail during flight. Using TReK as a foundation of requirements (at this time TReK was still in it's infancy and learning what it needed), we bought two top of the line (233 MHz) dual processor workstations for flight support. As technology advanced, so did TReK's requirements. Fortunately, the first payload we flew had a very low data rate (two 1 pkt/sec GSE packets of maybe 20 to 40 measurement IDs each) and we effectively used the dual 233-processor workstations for that flight. At that time, we had another payload beginning to need resources as well. This time we bought three top of the line, dual processor (667 MHz) workstations. One was immediately taken for developmental work on the payload. The other two were used for remote operations testing and configuration. This payload contained a high data rate Payload Data Services System (PDSS) packet that required processing by our application beyond TReK. Coupling our application with TReK's proved to max the computer on data processing. As time and money has been available, and as new projects have come into being, we've done our best to purchase at least two of the highest dual processing computers possible (given budget constraints) in order to support the payloads. Additionally we need computers for testing at KSC for PTCS. Considering we utilize our whole configuration when processing data and it is fairly tightly woven together, we usually have to carry one high-powered workstation to KSC and some sort of database server machine.

Summary

In short, we usually come out needing **1** TReK class computer for testing purposes (be it ScS testing prior to PTCS, PTCS, parallel flight testing, or other use), **2** TReK class computers (preferably of the highest processing speed/memory/disk space as able to afford), **2** moderately powered single processing computer with a high quality sound card for the Internet Voice Distribution System (IVoDS)/IVoDS backup/parallel use, **1** low powered "non-mission critical" computer for general web access, email access, and MPV. We were fortunate to utilize some spare computers that were surpluses for the "non-mission critical" computers.

Justification

One may ask why a remote payload utilizes such a large number of computers. I'll try to layout our reasons for each. One of the TReK class computers is utilized to process all of the payload's data. Depending on the requirements of the payload, this could range from 1 to 1000 packets per second. Not only does TReK have to process this data, but then our application must process and distribute this information. We elected to store the data to a database (and I'm getting ahead of this oversight, so more will be mentioned later on this). Also distribute payload health and status to other applications that are not tied to this particular computer, so it can be viewed from external locations (e.g., employees' offices) instead of having to dedicated constant time in our operations control center. This supported after hours checking from home for console personnel during weekend, or late night errors/mishaps. In general, the processing done by our in-house applications and TReK usually consume the vast majority of resources. In an effort to prevent loss of packets and data, we have dedicated one computer for that sole purpose. One additional computer, which can be used as a short-term backup for the data processing computer, is used for EHS access and TReK API Commanding, if required. In the course of daily operations, this computer tends to consume a large amount of screen real estate, since several different electronic documents and a variety of other screens have to be open and active in order to coordinate tasks. In an effort to prevent screen over-crowding, we try to minimize the additional applications running on this computer, though several applications could be added to this computer. The third computer we utilize is a low-end "non-mission critical" computer for use of the Manual Procedures Viewer (MPV). Due to a memory leak that could cause potential lock up of the computer, we've been advised to run MPV on a "non-mission critical" computer. Since we didn't have such a computer to start with, we had to get one. This computer now includes email for the mission as well as general web access. Due to the CPU requirements of TReK, the screen real estate consumption of the EHS computer, and the fact that IVoDS is a mission critical component, we had to find a fourth machine to handle IVoDS. While IVoDS could potentially be operated with other applications, it was believed best for ease of use, to put this application on a dedicated computer. This allows users to see and access the configuration, click-to-talk, and perform any other task with the voice communications, without having to sort through a variety of other windows.

Data Processing

MSFC provides no easy interactive storage system for the variety of needed payload, which gives them easy access to manipulating it into a desired format for processing, redistribution, reports, and/or summaries. Our application, we term a "proxy", provides that interface, but we needed to have the hardware available to store that data. Data requirements vary depending on the payload. Some payloads require only a small portion of the data that usually can be archived up to six months after flight. Other payloads are still accessing and interacting with data a year after their flight. Some amount of resources had to be acquired, configured, and maintained to handle that data storage. We have managed over the course of several missions and utilizing an add-on ability to acquire a DELL PowerVault with a respectable amount of storage. This database configuration allows us to store a flight's data in an unprocessed format, perform operations on it to refine it, and redistribute it in a variety of ways. Additionally this gives us a fine tuned monitoring point of data, so email notifications that result in pages can occur to give console operators notice of any anomalous data for near real time handling.

Personnel Costs

The next underestimate was that the majority of the work is in setting it up and then it runs by itself. The computers handling flight operations, database, and the computers handling testing all require administration. Someone has to manage this small network of computers. That takes a considerable amount of time. Security, configuration, testing, flight preparation, and a variety of other activities requires someone to spend time getting the computers needed configured appropriately for each task. Additionally, someone has to know how to install, configure and maintain the variety of NASA provided software applications on these machines.

Commercial Off-The-Shelf (COTS) Software purchasing

Another cost overrun we experienced is associated with COTS in general. Starting out, we were given a variety of COTS needed to use to interface with MSFC. The best example of all of this is, though not the only, is Exceed, an x-windows terminal emulator. When we started, Exceed had just released version 6.0 on the market. Documentation at MSFC stated to use version 5.2 of Exceed. We inquired if we were to use version 5.2 or the latest version 6.0. We were instructed to use the version stated on the requirements web page. The web page stating what software to purchase stated 5.2. Since this was an old version, we assumed that it wouldn't be changing. Shortly afterwards we're told to get version 6.0, then 6.1, then a patched version, and so forth. Finally, x-windows login for remote sites was phased out and migrated to the web. All the purchasing of the COTS that was done to provide that interface is no longer used. While one can argue that this cost was not wasted, since it was used for two of our flight missions, it is something I'd consider an unexpected cost due to the COTS short-lived use. Accessing MSFC has become less burdensome on the RPI since then. I don't believe we utilize any COTS beyond those that would be used in normal use (Microsoft Office, Windows 2000).

Data Access

Oftentimes we know what data we're going to receive in general, but nobody decides how they'd like to look at it until the last minute. We have to provide time and personnel to put together web applications and other tools that retrieve the data in the requested form. Console operators, engineers, and scientists all need to see different data at different levels of detail. This is often one of the last tasks addressed by a payload. With the aggressive schedule for development and testing, it usually is unknown until the payload itself is fairly refined and all involved get ample time to deal with the data. Additionally, not all of those who must be involved are involved at this stage. Scientists are often preparing the experiment ingredients and seldom are involved in the engineering aspects; engineers are seldom involved in the science aspects. The console operators come from a medley of disciplines and projects that may, or may not, be intimately involved with the payload, let alone with the data.

Training

Two approaches can be taken more or less with console operations.

1. A few people working constantly on the payload operations and not work on anything else, or
2. a large number of people each work a bit on payload operations, so everyone can do their daily work.

Regardless of which choice is made, people must be trained to use the applications MSFC has provided and understand how to relate that to their duties. Initial training was not an unexpected cost. What was unexpected was consistent retraining on both changed applications and unchanged applications. We were not expecting the application interfaces to constantly change. At one time it seemed every major build of EHS redesigned some major functionality interface. This causes confusion among console operators who a few months ago were accessing an application through one set of menus to have to access it a different way. Some users were quick to pick up on applications as well. Others take time to use and get familiar with the applications. When dealing with a large team, simulations months in advance from flight, and someone who needs repetition to reinforce learning, it is difficult to give everyone the training needed to the level that each need it. A wide variety of handbooks that detail the payloads level of interaction with various applications, both internal and external, have to be created and maintained as a tool for console operators to access and use. Training also often takes place months before flight. Delays in launch activities cause applications users to lose their knowledge quickly. Imagine being trained on how to use an application, having a simulation or two where the application is used, and waiting one to three months before you see the application again and then having to use it for flight operations. That is a very difficult case to handle.

I hope my explanation of some of the problems that have caused unexpected costs to us might help someone else keep from encountering those same costs. While at times frustrating, the ground systems as a whole have stabilized much compared to when payload operations first began. Most of these problems come more from growing pains than anything else. The task of running an ISS payload can seem daunting at times, but it is a manageable task.